This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

Beyond Reproducibility: Token Probabilities Expose Large Language Model Nondeterminism

Tairan Fu, Gonzalo Martínez, Javier Conde, Carlos Arriaga, Xiuyuan Qi, Shanshan Liu and Pedro Reviriego

Abstract—The execution of Large Language Models (LLMs) on GPUs has been shown to exhibit nondeterministic behavior, even under deterministic settings, due to finite-precision arithmetic and the order of parallel operations. This poses challenges for deploying LLMs in dependable systems, because the lack of consistency represents a fundamental reliability issue that complicates for example the auditability and certification of LLM decision-making. Previous studies mainly focus on output variability or enforcing deterministic execution. To enable a deeper analysis, this work investigates the nondeterminism at the token probability level. Through extensive evaluations across multiple LLMs, GPU platforms, and configurations, consistent results in both the trends and quantitative variations of probabilities are observed: nondeterminism significantly affects token probabilities in the range of 0.1 to 0.9, while probabilities near 0 or 1 remain relatively stable. Based on the findings, significant implications and insights for understanding nondeterminism in LLMs are provided in this paper. i) Nondeterminism can meaningfully affect the generated text when the temperature is not zero by perturbing mid-range token probabilities; ii) its impact depends on model characteristics such as probability distributions and output length; iii) the effect of nondeterminism may potentially be estimated using a single inference by analyzing token-level probabilities, without requiring repeated runs. Finally, the data and code produced in this work is released publicly to facilitate further investigation.

Index Terms—Artificial intelligence, Large language models (LLMs), Determinism

1 Introduction

The adoption of large language models (LLMs) has grown rapidly driven by their ability to perform a wide range of natural language processing tasks in artificial intelligence (AI), including text generation, summarization, translation, question answering, and code completion [1]–[3]. To perform inference, LLMs are commonly deployed on professional Graphic Processing Units (GPUs), supported by for example NVIDIA and Huawei, which provide massive parallelism and optimized hardware for the tensor operations achieving the high performance needed to generate text at reasonable speed [4], [5]. To maximize throughput, LLMs are often run with multiple concurrent inferences on the same GPU, a practice commonly referred to as batching [6], [7]. Batching improves utilization of GPU resources and reduces per-prompt overhead, enabling faster response times in production environments and making it a widely adopted strategy for handling high query volumes with efficient resource usage. When multiple prompts are processed together in a batch, the GPU executes operations across sequences in parallel. The exact order in which computations are carried out can vary depending on thread scheduling, memory access patterns, or kernel optimizations.

The evaluation of LLMs is essential for understanding their capabilities, limitations, and suitability for real-world applications. As these models are increasingly used in domains such as healthcare, education, finance, auditing, or legal verification, systematic assessment of metrics like accuracy, coherence, relevance, auditability and safety has become a central concern in AI research [8]. Without rigorous evaluation, apparent advances may mask subtle limitations in reasoning, factuality, or domain-specific understanding, hampering the effective and safe integration of LLMs into practical systems [9]. In these cases, reproducible results are required [10]. For example, consider a compliance system where the model is tasked with verifying whether a legal contract meets specific safety standards. A human auditor who submits the same contract to the system twice expects to get the same verdict.

To ensure reproducible results, it is commonly assumed that setting the temperature to zero and fixing random seeds is enough. However, recent research shows that LLMs can remain non-deterministic even in this case [11]. For example, [12] provides a large-scale empirical study of this phenomenon across different LLMs and tasks, demonstrating that nominally deterministic configurations can yield accuracy variations of up to 15% across identical runs, with performance gaps reaching up to 70% between best and worst-case outputs. The differences are attributed to floating-point rounding errors [13] and non-deterministic kernel behaviors, especially in fused attention and normalization operations. The role of floating-point non-associativity as a fundamental source of nondeterminism in high-performance computing and deep learning applications was studied in [14], showing that mathematically identical computations can yield different results due to different orders in the operations. To improve reproducibility, the use of higher precision floating-point formats was proposed in [15], showing that

Tairan Fu is with the Department of Mechanical Engineering, Politecnico di Milano, 20156 Milano, Italy (e-mail: tairan.fu@polimi.it).
Gonzalo Martínez, Javier Conde, Carlos Arriaga, and Pedro Reviriego are with the Information Processing and Telecommunications Center, ETSI de Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain (e-mail: {gonzalo.martinez.ruizdearcaute, javier.conde.diaz, carlos.arriaga.prieto, pedro.reviriego}@upm.es).
Xiuyuan Qi and Shanshan Liu are with the University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xiuyuanqi@std.uestc.edu.cn; ssliu@uestc.edu.cn). Corresponding author: S. Liu.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

more stable results can be obtained when using them both for the parameters and arithmetic operations or only for the latter. However, this comes at the cost of, for example, doubling the memory because now each parameter needs 32 bits instead of 16 bits, and the arithmetic operations are also more complex.

Nondeterminism is not only an issue for reproducibility, it can also be exploited to extract information from LLM providers as demonstrated in [16]; this relies on deterministic output comparisons to detect if a model has been changed to a lower cost/precision version. Potentially, an attacker can use nondeterminism to extract information about the hardware, batch size, or load of the provider. In fact, nondeterminism can be more broadly considered as a shift in the model's estimated token probabilities that has many potential implications beyond reproducibility, e.g., affecting performance when the temperature is larger than zero.

Recent studies on the high-level impact of nondeterminism have primarily focused on observing differences in output sequences across multiple inference runs, but less attention has been paid to token-level probability distributions and their variability. In this paper, a more general approach to study the impact of nondeterminism is pursued by considering changes at the token-level probability rather than directly on the LLM generated text. Measuring the logits or log-probabilities of each token across repeated runs provides a more fine-grained view of nondeterminism, revealing effects that may not immediately result in a token change. Metrics such as the standard deviation and range of token probabilities variations across runs allow quantification of both typical fluctuations and extreme deviations, highlighting the tokens or probability regions that are most sensitive to nondeterministic effects. By examining these measures across token sequences, it is possible to identify sensitive regions in the probability distribution and assess their potential impact on text outputs or downstream task performance.

This approach complements previous works both on the low-level mechanisms that cause nondeterminism and on the induced changes in the generated text, providing additional information on the impact at the application level. In particular, this paper makes the following contributions:

1) The nondeterminism is characterized as an uncertainty in the token probabilities, and two corresponding metrics to measure it are proposed.
2) The effect of nondeterminism on different GPUs has been examined; especially, the results for Huawei GPUs are evaluated and reported in the technical literature for the first time.
3) The uncertainty in the token probabilities is measured for different models, configurations, and hardware platforms, when the models run on a single GPU.
4) Based on the evaluation results, the following key trends are obtained: i) at the token probability level, the impact of nondeterminism follows a consistent pattern across all the models and GPUs considered; ii) the impact is significant when the token probability is in the range of 0.1 to 0.9 while much smaller when the probability is close to 0 or 1; iii) when the probability is not close to 0 or 1, the impact is similar regardless of the value of probability (e.g., either 0.2 or 0.8).
5) The implications of all results are discussed to illustrate the benefits of conducting the analysis at the token probability level.
6) A public dataset containing the results is provided to facilitate further research.

The rest of the paper is organized as follows: section 2 summarizes the related work that has explored nondeterminism in LLMs; section 3 introduces the proposed metrics used for measuring variations in token probabilities due to nondeterminism; section 4 presents the methodology used in the evaluation, and the results are presented in 5. The limitations of our study are summarized in section 6 and the main insights and implications are discussed in section 7. Finally, the paper ends with the conclusion in section 8.

2 Related Work

Nondeterminism in LLMs has been recently studied, challenging the common assumption that setting the temperature to zero, fixing random seeds, and using single-batch inference to guarantee consistent outputs. Empirical studies have shown that even under these intended "deterministic" configurations, repeated runs of the same prompt can produce different outputs [12]. The consequences of such nondeterminism are important. At the token level, it can induce small differences in logits and probabilities, which may affect top-k token selections, sampled sequences, or subsequent decoding behavior when temperature is non-zero. Although individual variations are often small, their effects can propagate through the generation process, particularly in long sequences, resulting in outputs that differ significantly in content, phrasing, or factual consistency [12].

Recent micro-architectural studies demonstrate that nondeterminism often originates from individual Transformer kernel operations. In particular, non-associative parallel reductions in LayerNorm computations and dynamic thread scheduling during fused attention passes, such as those in FlashAttention variants [17], introduce non-deterministic floating-point accumulation orders [18]. Atomic additions across parallel GPU blocks mean that identical inputs can yield subtle logit perturbations at the kernel level.

The impact of data formats has also been investigated regarding the nondeterminism issue. Commonly, several floating-point formats are used to run LLMs, including traditional FP16 and FP32, defined in the IEEE-754 Standard [19] as half and single precision formats with 16 and 32 bits, respectively; the newer BF16 [20] is also commonly used for models running in commercial GPUs. There are also efforts to develop emerging floating-point formats with fewer bits to reduce the memory and complexity of arithmetic operations [21] and circuits that combine multiplication and addition to reduce rounding errors [22]. Previously, the role of floating-point non-associativity [13] as a fundamental source of nondeterminism in high-performance computing and deep learning has been explored in [14]. The study highlights that mathematically identical computations can produce different results depending on the order of operations, which directly impacts the reproducibility of deep learning models. Therefore, to mitigate such a reproducibility problem in LLMs, [15] investigates the use of higher precision floating-point formats, showing that FP32 arithmetic applied to pa-

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

rameters, computation, or both can reduce nondeterministic fluctuations compared to BF16 and FP16, albeit at the cost of increased memory and computational overhead.

To mitigate output variance, prior research has explored software and system-level enforcement of deterministic inference execution. These techniques typically rely on forcing deterministic CUDA algorithms, locking thread schedules, executing batch-invariant reduction trees, or utilizing verified speculative decoding schedules [23], [24]. While effective at ensuring exact bitwise reproducibility, these mitigations frequently impose severe throughput overheads and latency penalties during large-scale serving [23].

Beyond concerns about reproducibility, nondeterminism in LLMs can have significant security and auditing implications. As demonstrated in [16], nondeterministic outputs can be leveraged to detect subtle changes in model deployments, such as when a provider substitutes a higher-cost or higher-precision model with a lower-cost or lower-precision variant. By comparing outputs over repeated queries, an attacker or auditor may be able to infer differences in hardware configurations, batch sizes, or server load conditions, potentially exposing internal operational details of the LLM service. These findings suggest that nondeterminism is not merely a technical nuisance but could be exploited to extract sensitive information or detect optimization strategies applied by model providers. Therefore, the understanding and characterization of nondeterministic effects is important not only for ensuring scientific reproducibility, but also for security auditing, regulatory compliance, and trustworthiness in environments where LLM outputs are used in sensitive decision-making or deployed as application programming interfaces (APIs).

In addition to the works on nondeterminism in LLMs, several studies have examined stability, variability, and reproducibility in deep neural networks more broadly. For example, [14] shows that floating-point non-associativity and parallel GPU/kernel execution can cause significant variability across runs, even when code, hyperparameters, and random seeds are held constant. Similarly, [25] demonstrates that training the same convolutional neural network multiple times under identical settings can lead to substantial differences in learned parameters and outputs, especially when GPU operations are non-deterministic, and that using higher precision arithmetic reduces, but does not eliminate this variability. Finally, [26] investigates reproducibility in generative models in general beyond LLMs, showing that even under deterministic decoding strategies, reproducibility is not always guaranteed, and that consensus or multiple independent runs may be needed to verify results.

These findings demonstrate that deep learning nondeterminism originates from a multi-layered interaction between floating-point arithmetic, kernel-level operator implementations, and dynamic runtime execution conditions. However, existing literature primarily evaluates macro-level outcomes, such as end-to-end model reproducibility or final output text, leaving the impact of non-determinism on internal token-probability distributions largely unexplored. This paper provides a fine-grained analysis at the token-probability level. By characterizing how low-level numerical perturbations alter probability distributions under standard operating conditions this work establishes a diagnostic foundation for understanding how hardware and runtime-level non-determinism translates into actual text generation behavior.

Furthermore, non-determinism varies across hardware architectures due to differences in execution pipelines and memory hierarchies. While prior work focuses almost exclusively on NVIDIA GPUs, emerging accelerators like Huawei's Ascend series remain unexplored. This paper addresses this issue by providing the first evaluation of LLM non-determinism on Huawei GPUs.

3 Token Probability-based Nondeterminism Metrics

As introduced previously, most of the existing works have focused on the differences in the generated text due to nondeterminism when the LLMs are run with the temperature set to zero. However, this provides only indirect information on the variability of LLMs. Looking at the token probabilities, even when doing greedy decoding, can provide additional information. In fact, the changes in the generated text are a consequence of the changes in the token probabilities.

To measure how token probabilities change across different runs, we have to first select the tokens to evaluate at each inference step i. As models have vocabularies with hundreds of thousands of tokens, it seems reasonable to focus only on the tokens with the highest probabilities. In particular, we select the top T tokens $t_{i,j}$ at each inference step i. However, even when doing that, the top tokens may be different from run to run due precisely to nondeterminism. To resolve this discrepancy and establish a consistent evaluation baseline, we select the top T tokens from a reference inference run performed with a batch size of one and greedy decoding.

Let us now consider an inference step i, where we obtain the probability distribution over the evaluated tokens. Over N independent runs, the probability assigned to a token $t_{i,j}$ at inference step i and run r is denoted as $p_r(t_{i,j})$. Treating this as a random variable, we compute the sample standard deviation $\sigma_{i,j}$ across the N runs as follows:

$$\sigma_{i,j} = \sqrt{\frac{1}{N}\sum_{r=1}^{N}\left(p_r(t_{i,j}) - p_{\text{ref}}(t_{i,j})\right)^2}, \tag{1}$$

where $p_{\text{ref}}(t_{i,j})$ is the probability of token $t_{i,j}$ in the reference run. Ideally, it should be zero, which corresponds to every run exactly reproducing the deterministic reference.

While the standard deviation provides an aggregate view of variability, it is also highly informative to observe the extremes of the probability distributions. To capture this, we calculate the absolute range ($R_{i,j}$) of the probabilities by taking the difference between the maximum and minimum values observed across the N runs:

$$R_{i,j} = \max_{1\le r\le N} p_r(t_{i,j}) - \min_{1\le r\le N} p_r(t_{i,j}). \tag{2}$$

Together, the standard deviation $\sigma_{i,j}$ and the range $R_{i,j}$ provide a fine-grained quantification of the impact of nondeterminism at each individual inference step for every evaluated token. Finally, it should be noted that the computational effort of computing these metrics is negligible compared to that of model inference, and does not impact model performance in terms of speed.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

4 Evaluation Methodology

To evaluate nondeterminism, the models, hardware platforms, prompts and batch sizes, and the evaluation procedure need to be specified; this information is provided in Fig. 1. The subsections next discuss each part in detail.

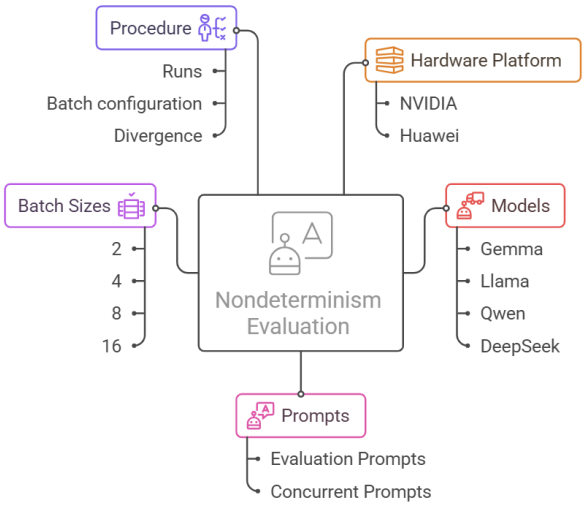


Fig. 1: Diagram showing the factors involved in nondeterminism evaluation.

4.1 Models

The evaluation focuses on LLMs that can run on a single GPU. The rationale for this decision is twofold. Firstly, this should be the best case for deterministic behavior, as running the models on several devices is likely to introduce additional variations during execution. Therefore, it provides a lower bound for nondeterminism. Secondly, it corresponds to a common scenario for local deployment of LLMs, making the results widely applicable.

From the models available at the time of writing this paper, we selected four LLMs with competitive performance from different companies so that the set is diverse and represents the state-of-the-art in models. Three are general models and the fourth is a reasoning model derived from another of the models evaluated. This enables us to evaluate whether reasoning has any significant impact on nondeterminism. Specifically, the models to be evaluated are the following.

1) Gemma3-12B: model from Google[1].
2) Llama3.2-11B model from Meta[2].
3) Qwen3-VL-8B model from Alibaba[3].
4) DeepSeek-Qwen3-8B model from DeepSeek[4].

4.2 Hardware platforms

For hardware platforms, industry-standard high performance GPUs are targeted. In particular, we consider configurations based on several NVIDIA GPUs: the A100 [27] data-center GPU, the consumer RTX-A6000 GPU [28], and the latest data-center GPU H200 [29]. We also evaluate alternative AI accelerators such as Huawei Ascend-910 [30] which is designed for AI workloads and also supports LLM inference[5].

4.3 Batch sizes

The batch size determines the number of input prompts that a model simultaneously processes during a forward pass. Larger batch sizes can increase throughput, but may affect memory usage and, in some cases, computation order [31]. The batch size is the factor that will induce nondeterminism when a model runs on a single GPU; this is valid because the prompt under evaluation will run concurrently with other prompts which can lead to different order in the operations. In our single-GPU setup greedy decoding is deterministic when the batch size is one, which as mentioned before, is used as the baseline to select the top T tokens for evaluation. To keep the computational effort reasonable, we use a logarithm scale to select batch sizes and consider the following values in the evaluation: 2, 4, 8, 16. These values should show the trends of nondeterminism with batch size if they exist.

4.4 Prompts

In the evaluation, we have two types of prompts:

1) **Evaluation prompts**: the prompts on which we will assess nondeterminism.
2) **Concurrent prompts**: the prompts that will run concurrently with the evaluation prompts when the batch size is larger than one.

Since the goal is to evaluate nondeterminism in realistic scenarios, we use a homogeneous set of prompts to load the system and also partly for the evaluation prompts. To do so, we take questions from the Massive Multitask Language Understanding (MMLU) dataset [32] which covers 57 diverse different categories ranging from algebra to world religions. This ensures that the questions are diverse and, at the same time, facilitates the reproducibility of our experiments and conducting additional experiments. In more detail, for each batch of size B, we randomly select $B-1$ questions from the more than 15,000 questions in MMLU and ask the model to answer the question providing an explanation. Therefore, the model answers questions on a wide range of topics that mimic the use of models in many real scenarios.

For the evaluation prompts, we also use randomly selected questions from the MMLU dataset as the ones used for the concurrent prompts. The idea is to test prompts that are similar to the rest of the prompts being executed in the system and that cover a wide range of topics.

4.5 Testing procedure

To evaluate nondeterminism, we set the batch size as B and construct batches using $B-1$ randomly selected MMLU questions and one evaluation prompt. In more detail:

1. https://huggingface.co/google/gemma-3-12b-it
2. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct
3. https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct
4. https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B

5. The experiments on the Huawei Ascend platform were conducted on an Ascend 910B2 device with 64 GB of HBM memory. The host system used a Kunpeng 920 CPU with 48 cores at 2.6 GHz and 220 GB of system memory. The software environment consisted of Ubuntu 22.04, Python 3.12, CANN 8.5.0, npu-smi 25.2.0, and PyTorch 2.6.0.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

1) Select 10 questions randomly from MMLU. These will be the set of 10 evaluation prompts.
2) Select $B-1$ questions from MMLU.
3) Run ten batches, each with one of the evaluation prompts and the $B-1$ prompts selected in 2).

The analysis is then conducted up to the point where at least one of the runs diverges, ensuring that all tokens included in the comparison were generated based on exactly the same preceding context. This approach guarantees a fair and consistent evaluation of token-level variations across runs. The evaluation is conducted for the $T = 10$ top tokens at each inference step.

To adequately characterize the distribution of non-deterministic effects in model outputs, we conduct N independent runs. The standard error of the estimated standard deviation is approximately as

$$SE_{\text{std}} \approx \frac{\sigma}{\sqrt{2N}}. \tag{3}$$

This expression indicates that the precision of the standard deviation scales with $1/\sqrt{N}$, independent of the magnitude of the underlying variability [33]. In addition, the observed range of token probabilities across runs provides a complementary measure of variability that captures the maximum spread of probabilities due to nondeterminism.

In all experiments, we set $N = 50$. With $N = 50$, the denominator factor $\sqrt{2N} = \sqrt{100} = 10$ places the standard error exactly one order of magnitude below the value of σ. Consequently, the measurement uncertainty is small, and large deviations observed in the token probabilities cannot be attributed to random experimental variation, confirming that our choice estimates the metrics with sufficient precision while offering a practical compromise between statistical reliability and computational cost. A post hoc nonparametric bootstrap over the $N = 50$ runs confirms this approximation empirically: the empirical standard error of $\hat{\sigma}$ exceeds the theoretical value by more than $1.5\times$ for only 3.5% of (heavy-tailed) tokens, and its relative uncertainty stays below 33% even in the worst case (median 9.3%), not affecting any of the order-of-magnitude comparisons reported.

5 Evaluation Results

In this section, the results of the experiments are presented and discussed. First, we look at the distributions of the ranges R_i and the standard deviations σ_i of the differences due to non-determinism, and of the token probabilities. Then, we analyze the results for each token probability value to understand the effects of the different parameters, such as the model, batch size, or hardware platform. Finally, an additional experiment is conducted to evaluate whether model size has any relevant effect on nondeterminism. The code and results of all the experiments are available in a public repository [6].

5.1 Distributions of token probability variations: range and standard deviation

The results in terms of the distribution of σ_i and R_i due to nondeterminism of the top-10 token probabilities across 50 runs for all generated tokens are shown in Fig. 2. It can be seen that in all the models evaluated, for most tokens the nondeterminism-induced variations are close to zero considering both metrics; specifically, approximately 90% tokens have a variation of less than 0.0001, while only roughly 1% of tokens reach a variation of 0.05. This seems to suggest that the effects of nondeterminism are low. It is also interesting to note that the trends and values are similar for all models both for range and standard deviation.

To better understand the token probabilities, in Fig. 3, we take a closer look at the top-10 token probability values by examining the distribution of their average probability value. It can be seen that most of the values, roughly 90%, are concentrated around zero (in the 0 to 0.1 range). This is because in many cases only one or at most a few tokens have sizable probabilities, while the rest are negligible. This means that on the top-10 tokens most have low values. In fact, for many token predictions, there is only a single token with relevant probability that is close to one and the rest are close to zero, for example, when a token completes or continues a word made of several tokens. These results combined with the previous distribution of ranges for the variations of token probability in Fig. 2 suggest that using top-p sampling [34] can mitigate the effects of nondeterminism, because it filters out tokens whose probability was close to zero (most of the tokens) and that have increased due to nondeterminism. However, as we will see next, that is not the case when examining the effect of nondeterminism versus token probability.

5.2 Analysis as a function of token probability

In Figs. 4 and 5 we analyze the dependence of the impact of nondeterminism on the token probability for different batch sizes, models, and GPUs in terms of the range R_i and standard deviations σ_i of the variations observed for each token. The results show consistent trends regardless of the model, GPU or batch size; the variation is negligible when the token probability is close to 0 or 1 and takes significant values when it is in the 0.2 to 0.8 range. Therefore, a first observation is that on our experiments, non-determinism is only an issue for tokens on which there are several candidate tokens with significant probabilities.

To better understand this behavior, we looked at the variations of the logits z_i before applying the softmax normalization to obtain the token probabilities [35]. The results show similar variations across all the token probability ranges. An example is shown in Fig. 6 for the A100 GPU. It can be seen that the values are similar for all probability values, including those close to 0 or 1. This suggests that softmax normalization of

$$p_i = \text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{4}$$

reduces the variations for tokens with probability close to 0 or 1 and increases the variations for the other tokens. Since softmax is a nonlinear and saturating function, its sensitivity to variations in z_i depends on the relative differences between logits. This explains the small probability variations when $p_i \approx 0$ or $p_i \approx 1$ and the large variations when p_i is in the mid range (e.g., $0.2 \leq p_i \leq 0.8$).

6. https://github.com/aMa2210/llm-nondeterminism/

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

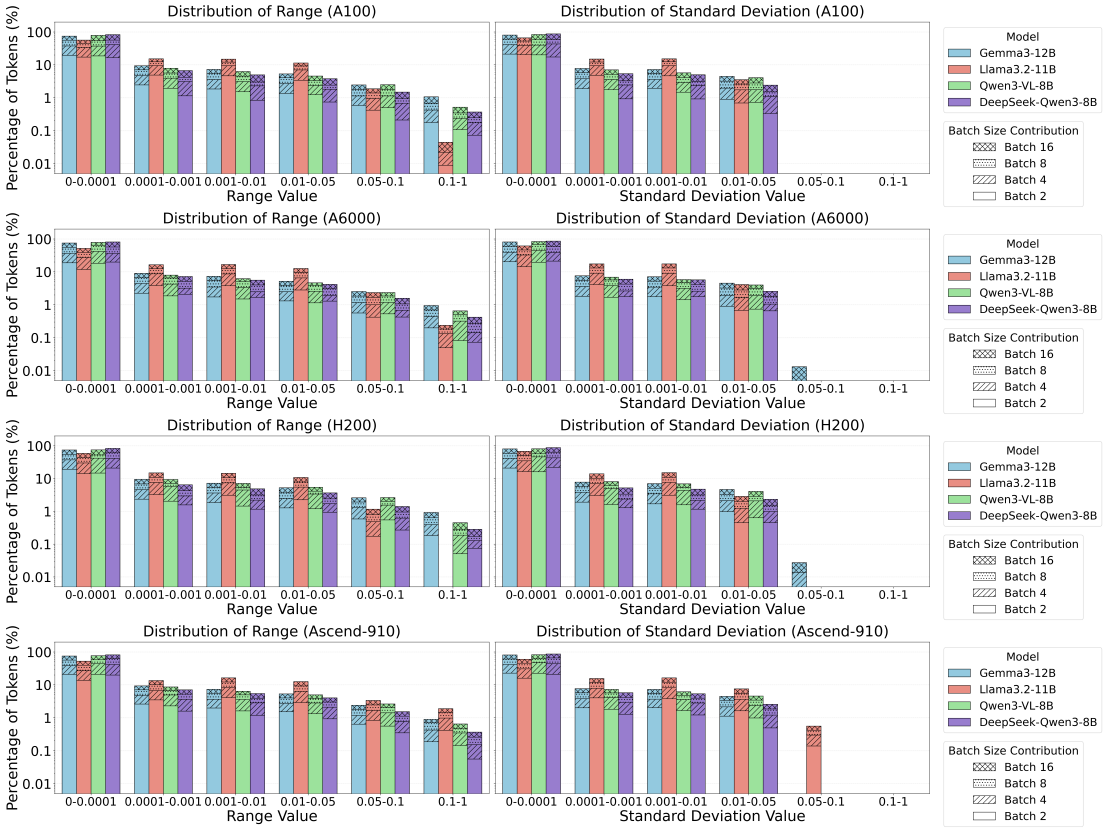


Fig. 2: Distribution of average range R_i and standard deviation σ_i of variations in token probability due to nondeterminism for different batch sizes and models across different GPUs.

Let us consider the different values of the probabilities:

Case 1: Probability close to 0. If a token has a probability close to 0, this implies that other tokens k have significantly larger logits, i.e., $z_k \gg z_i$. Therefore:

$$e^{z_i} \ll \sum_{j=1}^{n} e^{z_j} \approx \sum_{k \neq i} e^{z_k}, \tag{5}$$

and the softmax probability of Eq. (4) is close to zero and almost insensitive to small variations in z_i.

Case 2: Probability close to 1. If a token has a probability close to 1, this means that Eq. (6) and then (7) hold.

$$z_i \gg z_j \quad \forall j \neq i \tag{6}$$

$$p_i = \frac{e^{z_i}}{e^{z_i} + \sum_{j \neq i} e^{z_j}} \approx \frac{e^{z_i}}{e^{z_i}} = 1 \tag{7}$$

So, small perturbations δz_i result in:

$$p_i' = \frac{e^{z_i + \delta z_i}}{e^{z_i + \delta z_i} + \sum_{j \neq i} e^{z_j}} \approx 1, \tag{8}$$

and the probability stays close to one and changes very little.

Case 3: Probability in the mid range. When the token probability takes a moderate value (e.g., $0.2 \leq p_i \leq 0.8$), several logits are of comparable magnitude:

$$z_i \sim z_j \quad \text{for some } j \neq i. \tag{9}$$

Here, small changes δz_i can significantly shift the softmax probability because both the numerator and the denominator are of similar scale, leading to greater variations in the resulting probability. Therefore, the softmax function acts as a nonlinear amplifier in the central range and a suppressor near the boundaries. Fig. 7 represents this effect visually in the case of a two-token dictionary where the token probability (Eq. (10) matches the Sigmoid function of the difference of the logit values. It can be clearly observed how similar variations on the logits difference have a large impact on token probability in the middle probability range and a small impact when the probability is close to zero or one.

$$p_1 = \text{Softmax}([z_1, z_2])_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \tag{10}$$

This explains the observation that in our experiments nondeterminism is only an issue for tokens with several competing candidates of similar scores. This behavior is consistent across models, batch sizes, and GPUs because it is inherent to the mathematical properties of softmax rather than hardware specific effects.

Next, let us consider the impact of batch size. From Figs. 4 and 5 it can be seen that, both in range and in standard deviation, the variations increase with batch size for all models and GPUs. This trend suggests that larger batch sizes introduce greater variability in computations, potentially due to memory contention and differences in parallel execution patterns. Consequently, while increasing batch size can improve throughput and GPU utilization, it

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

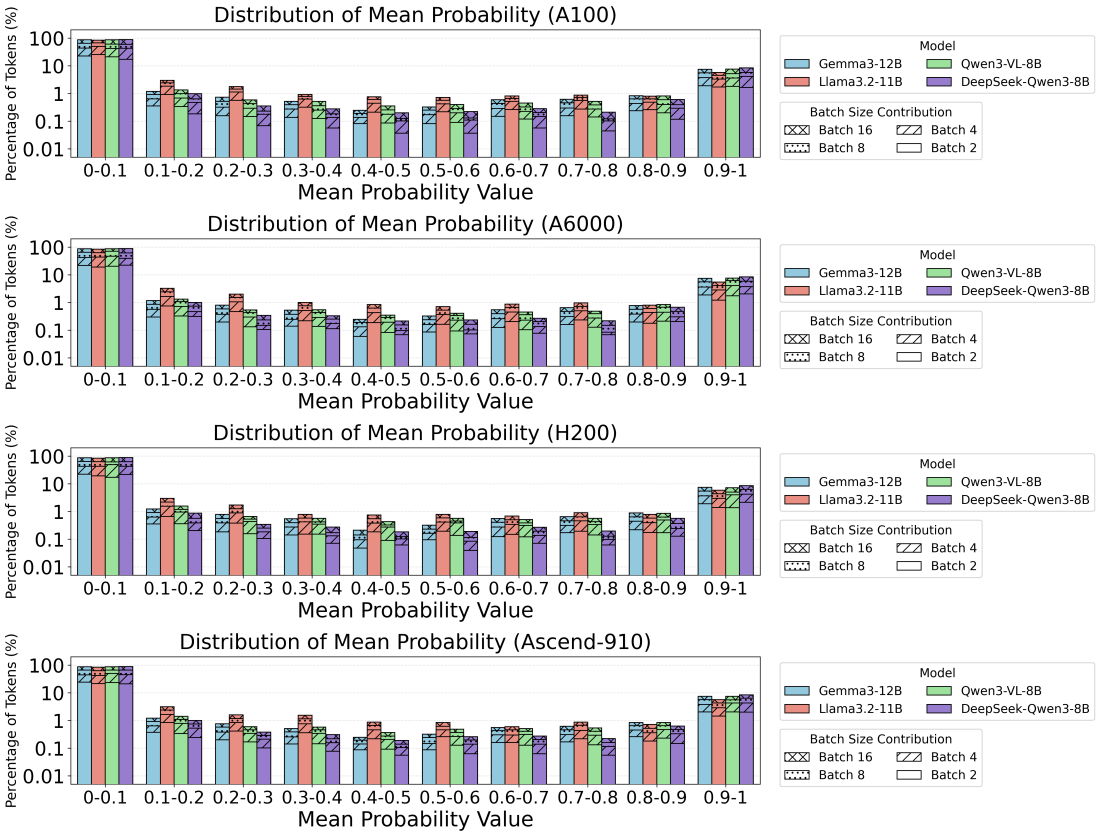


Fig. 3: Distribution of average token probability for different batch sizes and models across different GPUs.

also appears to amplify fluctuations in the results, which should be taken into account when selecting batch sizes.

When looking at the impact of models, the results for the four models evaluated are very similar, with Llama3.2-11B having slightly lower variations. This suggests that the impact of nondeterminism is comparable across open-weight models of similar scale, even when they come from different vendors. In other words, despite differences in architecture refinements or training data, models of roughly the same size appear to exhibit similar sensitivity to nondeterministic effects, with only minor deviations as those seen in Llama3.2-11B.

Finally, to compare the different hardware platforms, in Fig. 8 we plot the results for some of the models. It shows that both ranges and standard deviations are almost the same for the four GPUs. Therefore, it seems that the nondeterminism effects are roughly the same for given models and floating-point format in current GPUs. This is also an interesting result and suggests that the reordering of the operations due to parallel execution of prompts is similar in different implementations. Pairwise Kolmogorov–Smirnov tests on the σ_i distributions within each probability bin, model, and batch size formally support this similarity (median effect size $D = 0.14$; 92% of the 806 comparisons yield $D < 0.3$; complete results in our public repository).

5.3 Impact of Temperature

From our previous observations on how the softmax function influences nondeterminism, it seems that temperature can have two opposing effects. On one hand increasing the temperature puts more tokens on the mid probability range where nondeterminism is larger. However on the other hand, it smooths the softmax function so that the impact of logit differences on token probabilities are smaller. This is illustrated in Figure 9 which shows the proportion of tokens in the middle range (a) and the standard deviation of the tokens (b) due to non determinism for different temperatures.

The analysis of the impact of temperature on nondeterminism is left for future work and will likely depend on the trade-off between these two factors. While a higher temperature smooths the local impact of logit variations within specific probability bins, it simultaneously increases the total number of tokens exposed to the non-saturated zone.

5.4 Impact of model size

Another relevant parameter is the size of the model. To see if it has any effect on the level of nondeterminism, we evaluated four versions of the Gemma3 model with sizes of 270M, 1B, 4B and 12B, covering a range of close to two orders of magnitude in model size[7]. The models are from the same family to mitigate differences due to model architecture or design in an effort to concentrate on the impact of model size. The nondeterminism effects are similar for all sizes, with the same trends for both the range and standard deviation;

7. The models are from: https://huggingface.co/google/gemma-3-270m-it, https://huggingface.co/google/gemma-3-1b-it, https://huggingface.co/google/gemma-3-4b-it, https://huggingface.co/google/gemma-3-12b-it.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

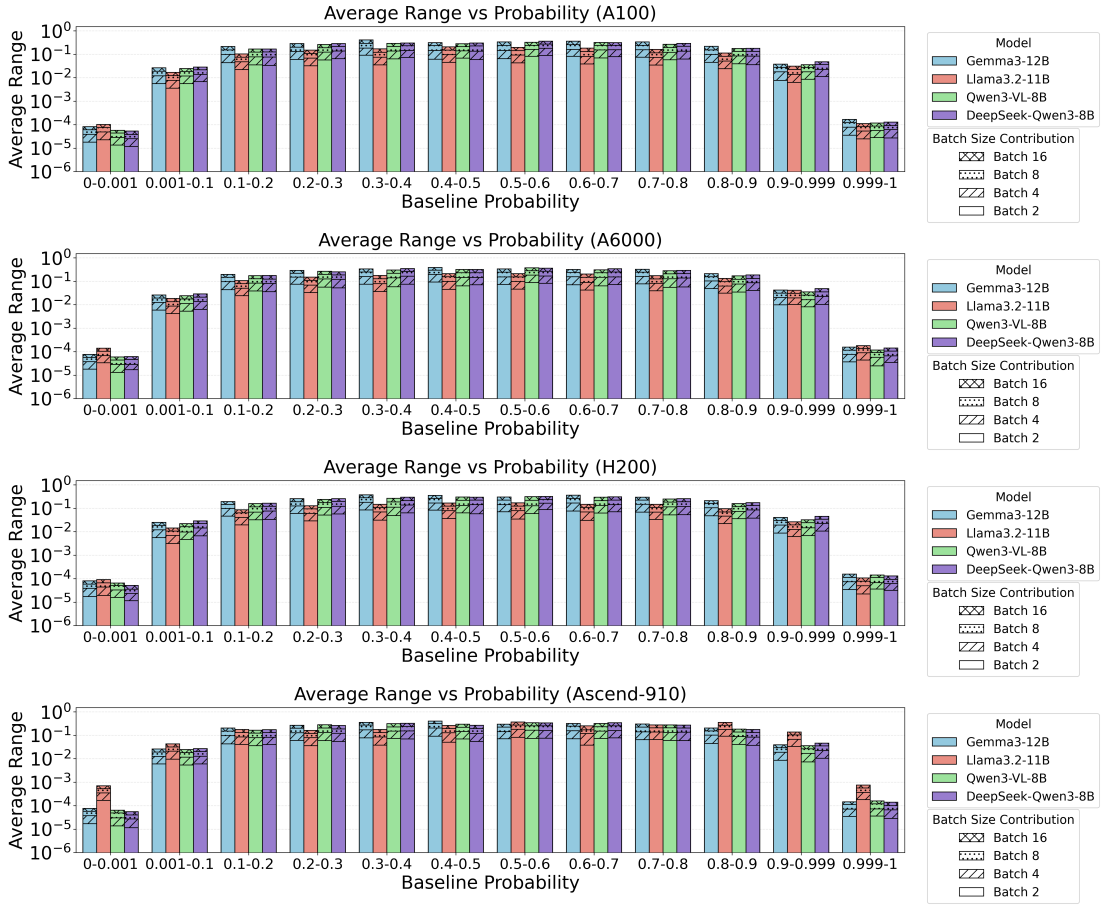


Fig. 4: Average range R_i of variations in token probability due to nondeterminism for different batch sizes and models as a function of the token probability for different GPUs.

specifically, they are smaller for probabilities close to 0 or 1 and have similar values for the rest of the probabilities. As an example, Fig. 10 shows the ranges. Interestingly, the absolute magnitude of these variations is also similar for the four model sizes, indicating that the model size does not significantly influence nondeterministic behavior in token probability outputs.

Finally, the runs of the smallest model, with only 270 million parameters, diverge almost immediately. For most questions, the generated sequences already differ at the very first token, with different runs opening their answers with different words (e.g., "The...", "Let's...", "Here..."). This is because the logit noise is large relative to the model's confidence margins, making tokens vulnerable to nondeterminism that can flip even the opening token. Since our analysis compares only the common prefix up to the first point of divergence, this leaves very few shared token positions to evaluate for this model, which is why several bars are missing from the figures.

5.5 Impact of Multi-GPU Configurations and APIs

Although our evaluation explicitly focuses on single-GPU configurations, we conduct an initial exploration to determine whether the observed non-determinism patterns appear in more complex execution environments.

To evaluate this, we design two distinct scaling experiments. First, we replicate a subset of our experiments using a multi-GPU setup consisting of a dual NVIDIA RTX A6000 configuration, enabling a direct structural comparison with our baseline single-GPU runs. Second, we expand our evaluation to a commercial, closed-source model (GPT-4o) via the OpenAI Application Programming Interface (API). While the precise underlying hardware cluster and parallelization strategy for the API are proprietary, it serves as an excellent proxy for highly complex, distributed multi-GPU serving infrastructure. The empirical findings for both setups are illustrated as a combined analysis in Fig. 11.

As shown in Fig. 11(a), moving from a single local GPU to a multi-GPU tensor or pipeline parallel setup ($2 \times$ A6000) yields similar average standard deviations (σ_i) across all token probability brackets. Similarly, Fig. 11(b) shows that the proprietary GPT-4o cloud infrastructure exhibits comparable non-deterministic behavior to a locally hosted Gemma3-12B model.

Therefore, these two experiments suggest that the mathematical transformation of logit fluctuations through the Softmax function modulates the impact of nondeterminism beyond single GPU implementations and may apply generally regardless of the scale or complexity of the hardware deployment. Further experiments on multi-GPU and closed models are needed to confirm this hypothesis

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

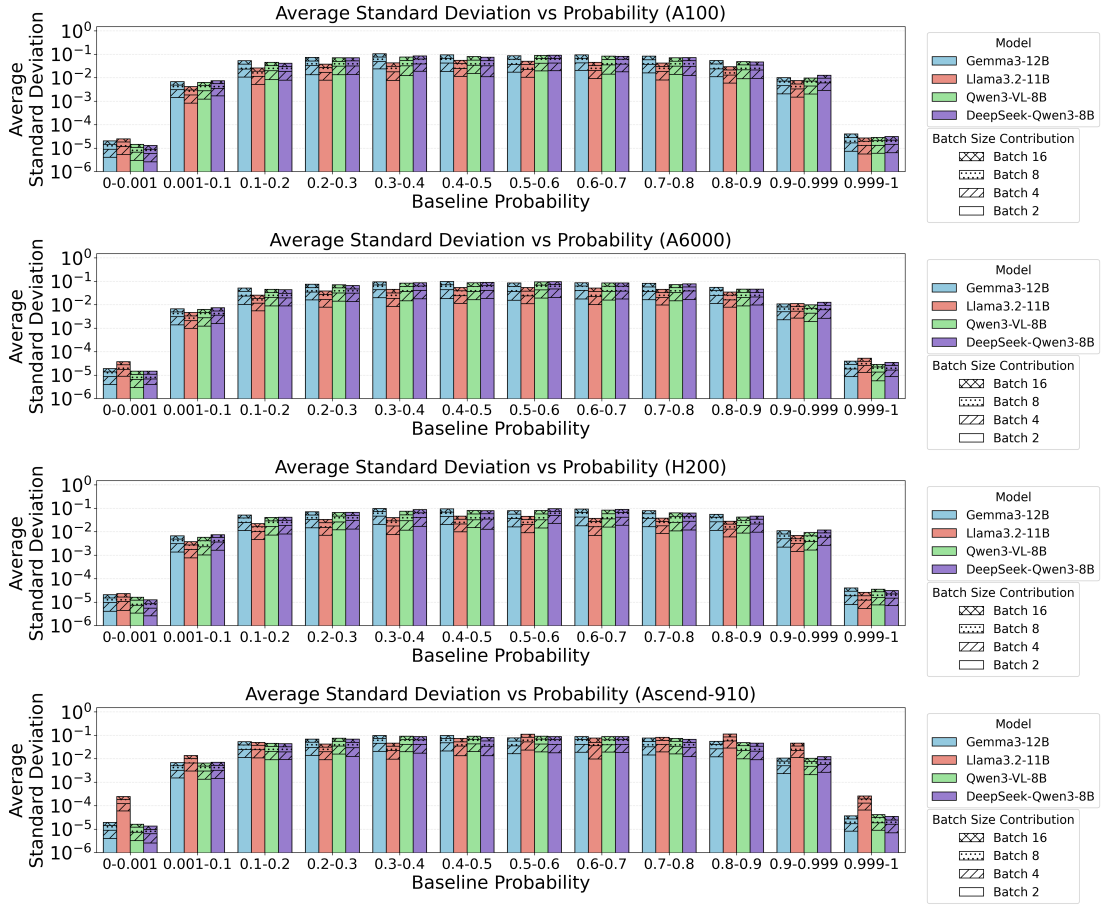


Fig. 5: Standard deviation σ_i of the variations in token probability due to nondeterminism for different batch sizes and models as a function of the token probability for different GPUs.

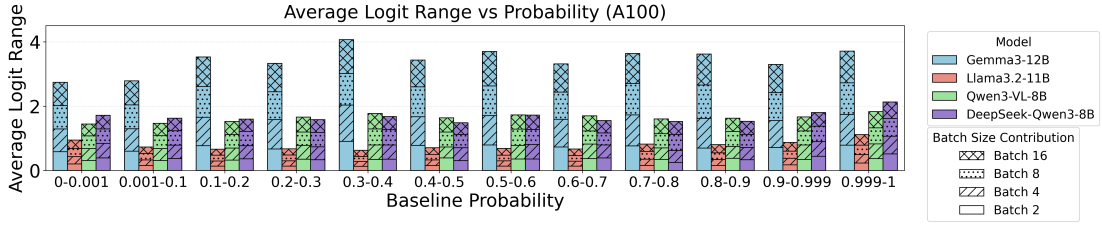


Fig. 6: Average range R_i of variations of the logits due to nondeterminism for different batch sizes and models as a function of the token probability for NVIDIA A100.

6 Limitations

Before discussing the main insights and implications of our work, we want to explicitly acknowledge the limitations of our study and how they have been mitigated when possible.

1) **Models**: only four open weight models of roughly the same size (8B to 12B) from different companies and four models of different sizes (270M to 12B) from a single company have been evaluated. These selections allow us to study nondeterminism effects across both similar sized models and models within the same family, but the coverage remains limited. Although this has been partly mitigated by conducting an initial evaluation of an OpenAI model, additional vendors and a broader range of model sizes should be evaluated to confirm whether the trends observed here are consistent and generally applicable.
2) **Single GPU deployment**: experiments were run on a single GPU. Nondeterministic effects may differ when a model is executed across multiple GPUs, where additional sources of variation, such as inter-GPU communication, partitioning, or synchronization can influence the variations in token probability. This has been partly mitigated by running a subset of the experiments on a dual A6000 configuration.
3) **GPU vendors and models**: four different GPU models from two vendors were used in the evaluation. While this provides some diversity in hardware configurations, additional GPU vendors and architectures should be evaluated to determine whether the nondeterminism effects observed are consistent across a wider range of hardware platforms. To address this issue, targeted experiments have been run on a More Threads, MTT S4000 GPU that also confirm the observations that tokens with midrange probabilities are affected the most by nondeterminism. The results and plots are available in the public repository.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

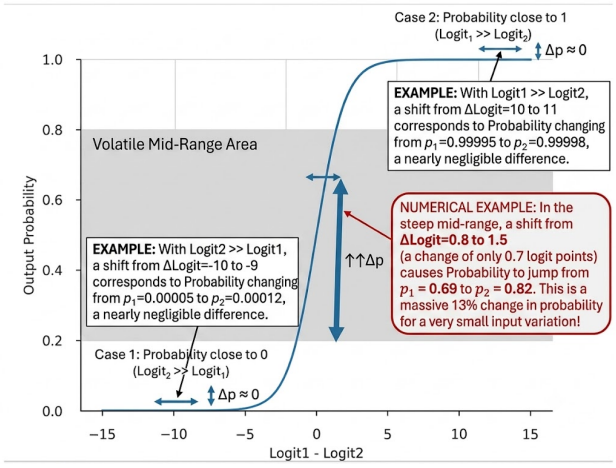


Fig. 7: Effect of Softmax in nondeterminism. The probability of the first token in a two-token vocabulary calculated by Softmax is identical to the probability of the positive class calculated by the Sigmoid function on the difference of the logits.

4) **Prompts**: Although prompts covering different topics were used by randomly selecting questions from the MMLU benchmark for the evaluation and concurrent prompts, additional testing with a broader variety of prompt types—including, for example, longer instructions, conversational inputs, creative tasks, and domain specific prompts, would help assess whether the observed nondeterminism patterns apply in general. To explore how nondeterminism depends on the prompt or task, we have run a set of prompts from the Infinity-Chat benchmark composed of open questions designed to evaluate the creativity of models [36]. As shown in Fig. 12, the resulting standard deviation profile for the open-ended Infinity-Chat benchmark closely mimics the parabolic distribution observed on MMLU. The unconstrained conversational inputs from Infinity-Chat exhibit a systematically higher variance floor across the volatile mid-range bins compared to multiple-choice tasks.
5) **Precision format**: the results were obtained using BF16, which is commonly employed for LLM inference. However, other numerical formats (such as FP16 or FP32) should be evaluated to determine the influences of choosing different precision formats on nondeterminisim. An initial experiment has been run with FP16 and FP32 on two different GPUs (A100, MTT S4000) to address this issue. The results show similar trends, the impact of nondeterminism concentrates on mid-range probability tokens. The results and plots are available in the public repository.
6) **Temperature**: the softmax used to compute token probabilities was applied with a temperature of 1, ensuring that no additional scaling was introduced during probability normalization. However, to address this limitation an initial exploration was done with lower and higher temperatures showing that temperature affects the fraction of tokens that fall in the mid probability range and also modulate the translation of logit variations into probability variations. A full analysis is left for future work.
7) **Runs**: each configuration is executed 50 times. This number seems sufficient to reveal the main variations, but performing a larger number of runs could potentially increase the observed range of values or alter the calculated standard deviation.
8) **Language**: all prompts used in the experiments were written in English. It would be valuable to extend the analysis to other human languages or even programming languages, to assess whether nondeterminism exhibits similar behaviors across different linguistic and structural domains.
9) **Downstream impact**: the connection to end-to-end task performance remained to be quantified. To address this, we conducted a comparative analysis using 10 random MMLU questions at a temperature of $T = 1$. By fixing the sampling seed, we established a deterministic baseline (batch size 1) where responses were identical across all 10 runs. In contrast, running the same prompts in a nondeterministic setting (single GPU with batch size 16) resulted in variations in more than 10% of the generated responses. These results confirm that token-level nondeterminism directly manifests as functional variability in downstream generation tasks as reported in previous works [12].

7 Discussion

In this work, we have, to our knowledge, analyzed in detail the variation introduced by nondeterminism to token probabilities for the first time. This effect impacts model performance in ways that go beyond reproducibility, which has traditionally been the primary concern regarding nondeterminism. The probability variations can subtly alter the model's behavior during text generation, influencing the tokens to be selected and ultimately affecting the quality and consistency of the generated text.

Although other factors and parameters such as the softmax temperature T introduce variability in token probabilities, their effects are fundamentally different from nondeterminism. Changing T systematically scales the probabilities without changing their relative order, making higher probability tokens still remain more likely. In contrast, nondeterminism introduces random fluctuations that can alter the order of token probabilities, especially for intermediate probability tokens, potentially changing the token sampled in a given run. As a result, nondeterminism introduces stochastic noise that can unpredictably modify the model's output. In the following subsections we first discuss the insight gained from the token probability analysis and then the implications of our study.

7.1 Insight beyond text-level nondeterminism studies

As reviewed in section 2 prior work typically quantifies nondeterminism mainly at the text level (e.g., whether repeated runs at temperature 0 produce different decoded tokens or sequences). Compared to it the token-probability analysis provided in this paper offers the following additional insights that are not accessible from text-only comparisons:

1) **Non-Determinism Depends Strongly on Baseline Token Probability:** Text-level studies can only report

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

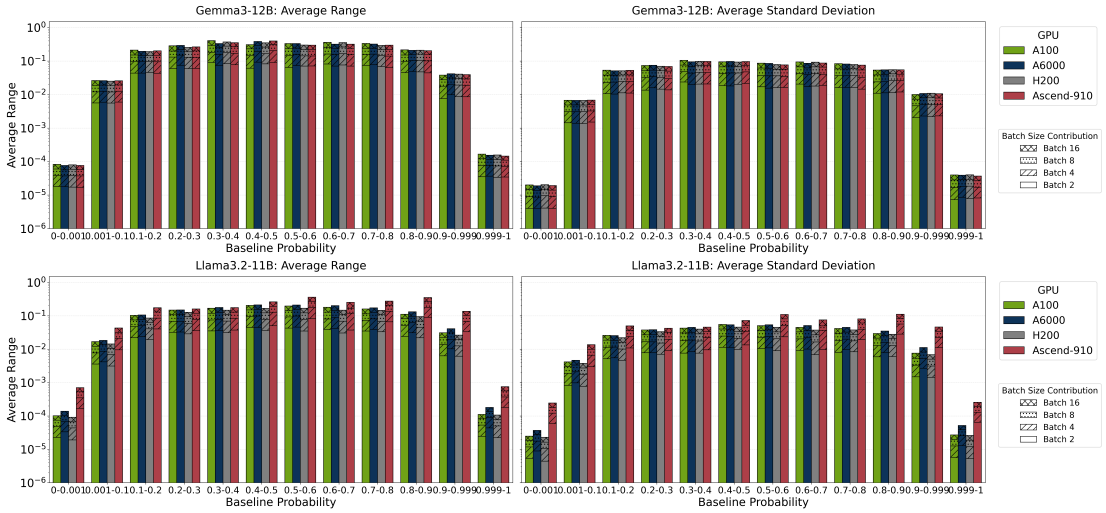
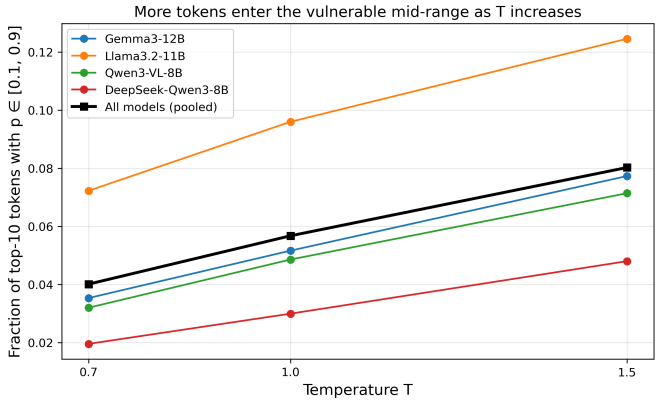
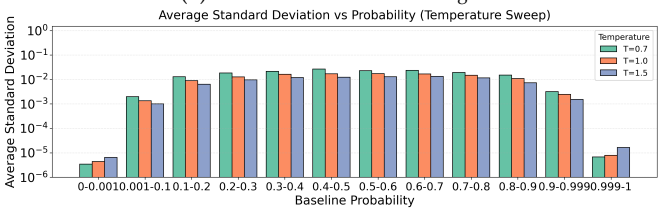


Fig. 8: Standard deviation σ_i and range R_i of the variations in token probability due to nondeterminism for different batch sizes and GPUs as a function of the token probability for different LLMs.



(a) Fraction of tokens in mid-range.



(b) Average standard deviation vs. probability.

Fig. 9: The dual impact of temperature on non-determinism across models: (a) higher T increases structural exposure by pushing more tokens into the vulnerable mid-range ($p \in [0.1, 0.9]$), while (b) higher T simultaneously smooths and attenuates the local noise magnitude (σ_i) within individual mid-range probability bins.

that a sequence flipped, but cannot predict when or why. Our distribution analysis proves that hardware-induced non-determinism is not random uniform noise. Instead, the expression of variance is conditionally bounded by the baseline token probability.

2) **The Volatile Mid-Range is Most Vulnerable to Hardware Jitter:** By inspecting the underlying distribution space, we demonstrate that structural sensitivity follows a distinct parabolic trajectory. Tokens residing at the boundaries ($p \to 0$ or $p \to 1$) are almost not affected by nondeterminism. Conversely, tokens situated in the mid-range ($p \in [0.1, 0.9]$) exhibit larger changes in output probability.
3) **Cross-Architecture and Cross-Model Structural Invariance:** The shape, peak boundaries, and noise floors of the empirical standard deviation (σ_i) profiles are similar across the four single-GPU setups and four open-weights models evaluated and also on the initial multi-GPU configuration, and proprietary cloud API model tested.

7.2 Practical Implications

The insights can be used to provide actionable outputs with practical implications, for example, the ones detailed below:

1) **Detect hidden nondeterminism even when the text is unchanged:** Greedy decoding can yield identical tokens across runs while the underlying probability mass assigned to competing candidates varies, especially in the mid-probability regime (approximately 0.1 to 0.9). However, traditional text-level metrics report zero variance in such cases. This phenomenon is illustrated in Fig. 13 as an example: in the top row, many generation steps are near-saturated (top token $p \approx 1$), whereas in the bottom row several steps show clear competition among candidates. Text-level metrics would treat both cases as identical if the argmax token does not change, whereas the visualized distributions reveal underlying model fragility.
2) **Enable finer and more actionable diagnosis:** By localizing the decoding steps that operate in the "fragile" mid-probability regime (where nondeterminism has the largest effect on the Softmax derivative), token-level metrics provide a direct way to identify prompts/steps most likely to exhibit run-to-run variability, providing a visual diagnostic tool while text-level agreement alone cannot isolate it.
3) **Explaining differences across models via probability regimes and length:** In our experiments, different models exhibit similar levels of non-deterministic variation at the token-probability level. Variations in generated-text performance (e.g., benchmark accuracy fluctuations) may be driven not by differences in nondeterminism per se, but rather by differences in (i) the model's underlying

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

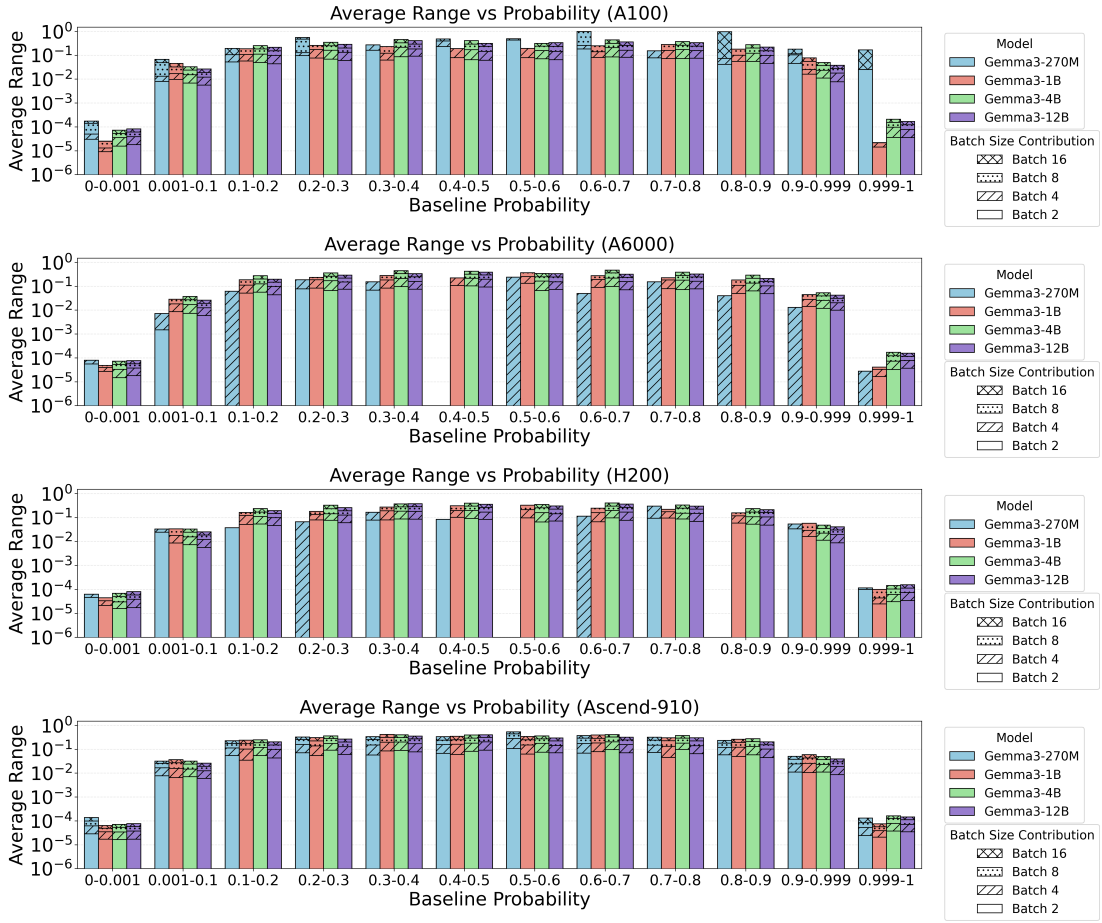
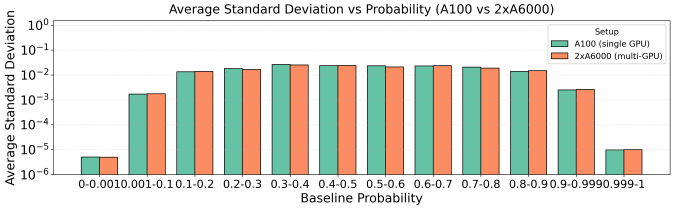
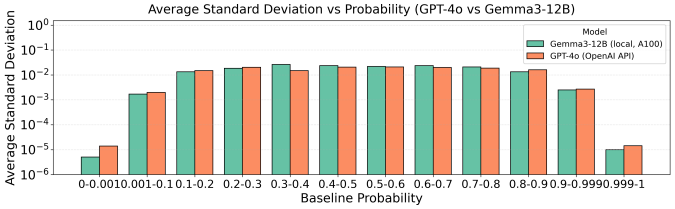


Fig. 10: Average Range R_i of variations due to nondeterminism for different batch sizes and models as a function of the token probability for different GPUs.



(a) Single-GPU vs. Multi-GPU execution.



(b) Local model vs. Closed-source API.

Fig. 11: Empirical standard deviation σ_i across alternative deployment architectures on MMLU ($T = 0$): (a) contrasting single-GPU (A100) and multi-GPU ($2 \times$ A6000) configurations for Gemma3-12B, and (b) comparing local open-weights execution with the closed-source GPT-4o API. Both plots confirm that the same profile of non-determinism remains invariant to hardware scale and API abstractions.

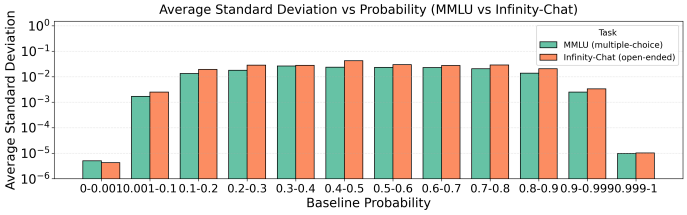


Fig. 12: Empirical standard deviation σ_i across task structures for Gemma3-12B on an A100 GPU: comparing highly constrained multiple-choice formatting (MMLU) against free-form, conversational creative tasks (Infinity-Chat). While open-ended dialogue exhibits slightly heightened sensitivity across the mid-range, both setups follow the identical parabolic trend dictated by the Softmax transformation.

token-probability distribution characteristics (models with a higher proportion of probabilities near 0 or 1 like the top row of Fig. 13 are inherently robust to nondeterminism) and/or (ii) average response length (shorter answers expose fewer tokens to stochastic variation).

4) **Potential single-run estimate of nondeterminism sensitivity:** Our findings suggest that a methodology could be developed to estimate the potential impact of nondeterminism from a single inference run, bypassing the need for computationally expensive repeated executions. This can be done at two levels of detail. In the first, we can just look at the fraction of tokens in the generated sequence that have probabilities in the middle range (0.1 to 0.9) and thus have stronger nondeterminism. The larger the fraction, the greater the sensitivity. This is clearly seen in the example of Fig. 13 with the color mapping assigned to the token probabilities. The second is to develop algorithms to provide more detailed estimations of the variations introduced by nondeterminism. To do so, we can for example, first compute the standard deviation across N runs for a few reference prompts under the same hardware and parameters configuration.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

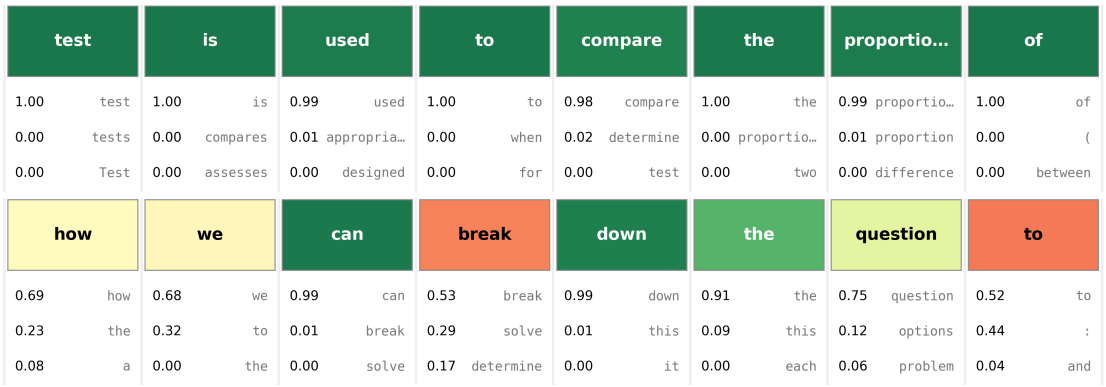


Fig. 13: Example of token-level probability visualization (generated by Gemma3-12B with batch size of 16). Each column corresponds to a generation step and lists the top three candidate tokens with their probabilities. Steps with saturated probabilities ($p \approx 1$ or $p \approx 0$) marked in green are typically stable, while steps with multiple plausible candidates (mid-range probabilities) marked in yellow/orange are more sensitive to nondeterminism.

Then, for a generated text sequence on a single run, we can estimate the non-determinism of each token using the standard deviation obtained in the reference prompts for that probability range. For example, if a generated token has a probability of 0.16, then its variation due to non-determinism will be approximated by the average standard deviation of the tokens in the reference prompts with probabilities in the range 0.1 to 0.2. As an initial experiment, we have applied this simple method using nine of the prompts as a reference and the other one to test the scheme, and repeated the process for all ten prompts. The results show a mean relative absolute error of only 30% in predicting the standard deviation of nondeterminism variations. This shows the potential of developing nondeterminism prediction schemes based on token probability information; however, the exploration of such a methodology is left as a subject for future investigation.

8 Conclusion

Prior research on investigating the effects of nondeterminism of LLM inference has focused primarily on the reproducibility of generated text. In this work, we took a closer look and conducted the first systematic analysis of nondeterminism in LLMs at the token probability level, rather than solely at the text output level. By evaluating multiple GPU platforms, batch sizes, prompts, models, and run conditions, we demonstrate that probability level of nondeterminism can significantly affect generated text when the temperature is greater than zero. We further observe that different models exhibit similar levels of nondeterministic variation at the probability level, with negligible impact for probabilities close to 0 or 1 and significant impact for probabilities in the range of 0.2 to 0.8. Hence, differences in benchmark performance across models may be partly due to differences in their underlying probability distributions or response lengths and not only to differences in nondeterminism itself. Models that produce probabilities concentrated near 0 or 1 are naturally less sensitive to nondeterministic effects, and models that generate shorter answers are affected less simply because fewer tokens are exposed to stochastic variations.

Building on these findings, several directions open for future research. One is to extend the analysis across a broader set of GPUs, models and prompts. A deeper theoretical understanding of the accumulation of probability perturbations across layers of the models would also help clarify their impact. Exploring mitigation strategies such as hybrid precision computation based on token probabilities could provide practical solutions to improve stability in real-world deployments. Finally, the design and evaluation of nondeterminism estimators from a single run probabilities such as the ones discussed in section 7 would also be of interest.

Acknowledgments

This work is supported by the FUN4DATE (PID2022-136684OB-C22) project funded by the Spanish Agencia Estatal de Investigación (AEI) 10.13039/501100011033, by TUCAN6-CM (TEC-2024/COM-460), funded by CM (ORDEN 5696/2024), and by the BRIDGE-AI (Grant 101299050) project funded by the European Union's Horizon Europe research and innovation program.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[2] X. Lyu, J. Li, D. Wei, M. Zhang, S. Tao, H. Yang, and M. Zhang, "Multiphase and multitask prompt tuning for llm-based context-aware machine translation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2025.

[3] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[5] J. Cheng, H. Kang, Y. Shao, N. Li, P. Chen, R. Wang, S. Long, X. Yang, and L. Ma, "Survey on efficient large language models: Principles, algorithms, applications, and open issues," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2025.

[6] Y. Sheng, L. Zheng, B. Yuan, Z. Li, M. Ryabinin, D. Y. Fu, Z. Xie, B. Chen, C. Barrett, J. E. Gonzalez, P. Liang, C. Ré, I. Stoica, and C. Zhang, "Flexgen: High-throughput generative inference of large language models with a single gpu," *arXiv preprint arXiv:2303.06865*, 2023.

[7] J. Li, J. Xu, S. Huang, Y. Chen, W. Li, J. Liu, Y. Lian, J. Pan, L. Ding, H. Zhou, Y. Wang, and G. Dai, "Large language model inference acceleration: A comprehensive hardware perspective," *arXiv preprint arXiv:2410.04466*, 2024.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2026.3732142

[8] B. Zhang, M. Takeuchi, R. Kawahara, and et al., "Evaluating large language models with enterprise benchmarks," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)* (W. Chen, Y. Yang, M. Kachuee, and X.-Y. Fu, eds.), (Albuquerque, New Mexico), pp. 485–505, Association for Computational Linguistics, Apr. 2025.

[9] M. T. R. Laskar, S. Alqahtani, M. S. Bari, M. Rahman, M. A. M. Khan, H. Khan, I. Jahan, A. Bhuiyan, C. W. Tan, M. R. Parvez, E. Hoque, S. Joty, and J. Huang, "A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds.), (Miami, Florida, USA), pp. 13785–13816, Association for Computational Linguistics, Nov. 2024.

[10] S. Biderman, H. Schoelkopf, L. Sutawika, L. Gao, J. Tow, B. Abbasi, A. F. Aji, P. S. Ammanamanchi, S. Black, J. Clive, *et al.*, "Lessons from the trenches on reproducible evaluation of language models," *arXiv preprint arXiv:2405.14782*, 2024.

[11] R. E. Blackwell, J. Barry, and A. G. Cohn, "Towards reproducible llm evaluation: Quantifying uncertainty in llm benchmark scores," *arXiv preprint arXiv:2410.03492*, 2024.

[12] B. Atil, S. Aykent, A. Chittams, L. Fu, R. J. Passonneau, E. Radcliffe, G. R. Rajagopal, A. Sloan, T. Tudrej, F. Ture, *et al.*, "Non-determinism of" deterministic" llm settings," *arXiv preprint arXiv:2408.04667*, 2024.

[13] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM computing surveys (CSUR)*, vol. 23, no. 1, pp. 5–48, 1991.

[14] S. Shanmugavelu, M. Taillefumier, C. Culver, O. Hernandez, M. Coletti, and A. Sedova, "Impacts of floating-point non-associativity on reproducibility for hpc and deep learning applications," in *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 170–179, IEEE, 2024.

[15] J. Yuan, H. Li, X. Ding, W. Xie, Y.-J. Li, W. Zhao, K. Wan, J. Shi, X. Hu, and Z. Liu, "Give me fp32 or give me death? challenges and solutions for reproducible reasoning," *arXiv preprint arXiv:2506.09501*, 2025.

[16] W. Cai, T. Shi, X. Zhao, and D. Song, "Are you getting what you pay for? auditing model substitution in llm apis," *arXiv preprint arXiv:2504.04715*, 2025.

[17] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 16344–16359, 2022.

[18] A. Arun, U. Srivastava, Y. Zhang, and R. A. Popa, "Nondeterminism-Aware Optimistic Verification for Floating-Point Neural Networks," *arXiv preprint arXiv:2510.16028*, 2025.

[19] "IEEE standard for Floating-Point arithmetic," July 2019.

[20] I. Corporation, "Bfloat16 – hardware numerics definition," Tech. Rep. 338302-001US, Revision 1.0, Intel Corporation, November 2018.

[21] P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, *et al.*, "Fp8 formats for deep learning," *arXiv preprint arXiv:2209.05433*, 2022.

[22] F. Niknia, Z. Wang, S. Liu, P. Reviriego, Z. Gao, P. Montuschi, and F. Lombardi, "A configurable floating-point fused multiply-add design with mixed precision for ai accelerators," *IEEE Transactions on Circuits and Systems for Artificial Intelligence*, vol. 2, no. 3, pp. 248–261, 2025.

[23] P. Sun, L. Zheng, I. Stoica, and Y. Sheng, "LLM-42: Enabling Determinism in LLM Inference with Verified Speculation," *arXiv preprint arXiv:2601.17768*, 2026.

[24] H. Wang, B. Chen, and J. Zhao, "Deterministic Inference across Tensor Parallel Sizes That Eliminates Training–Inference Mismatch," *arXiv preprint arXiv:2511.17826*, 2026.

[25] W. G. Pinto, A. Alguacil, and M. Bauerheim, "On the reproducibility of fully convolutional neural networks for modeling time–space-evolving physical systems," *Data-Centric Engineering*, vol. 3, p. e19, 2022.

[26] E. Kim, I. Isozaki, N. Sirkin, and M. Robson, "Generative artificial intelligence reproducibility and consensus," *arXiv preprint arXiv:2307.01898*, 2023.

[27] NVIDIA-Corporation, *NVIDIA A100 Tensor Core GPU – Data Center Product Page*. Accessed 2025.

[28] NVIDIA-Corporation, *NVIDIA RTX A6000 Workstation GPU – Datasheet*. Accessed 2025.

[29] NVIDIA-Corporation, *NVIDIA H200 GPU – Quick Specs*. Accessed 2025.

[30] HiSilicon-Huawei, "Hisilicon ascend 910 ai processor – technical overview," 2019. Accessed via "HiSilicon (Ascend) – AI acceleration" on Wikipedia.

[31] F. Yu, H. Zhang, A. Chen, X. Wang, X. Liang, S. Wang, G. Li, H. Cui, and X. Feng, "Characterizing and understanding deep neural network batching systems on gpus," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 3, no. 4, p. 100151, 2023.

[32] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.

[33] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. Hoboken, NJ: Wiley, 6th ed., 2014.

[34] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[36] L. Jiang, Y. Chai, M. Li, M. Liu, R. Fok, N. Dziri, Y. Tsvetkov, M. Sap, and Y. Choi, "Artificial hivemind: The open-ended homogeneity of language models (and beyond)," *Advances in Neural Information Processing Systems*, vol. 38, 2026.

BIOGRAPHIES

Tairan Fu is a PhD student at Politecnico di Milano, Milan, Italy. His research interests include Large Language Models, Diffusion Models, and Digital Twins.

Gonzalo Martínez is an assistant professor at the Universidad Politécnica de Madrid, Madrid, Spain. His research interests include Natural Language Processing, Security, Large Language Models y Artificial Intelligence.

Javier Conde is an assistant professor at the Universidad Politécnica de Madrid, Madrid, Spain. His research interests include Digital Twins, Big Data, Semantic Web, and Artificial Intelligence.

Carlos Arriaga is a Ph.D. candidate at Universidad Politécnica de Madrid, 28040, Madrid, Spain. His research interests includes videoconference and artificial intelligence. Arriaga received his Master's degree in engineering from Universidad Politécnica de Madrid.

Xiuyuan Qi received the B.S. degree in electronic information engineering from Harbin Engineering University, Harbin, China, in 2022. He is currently pursuing the Ph.D. degree with University of Electronic Science and Technology of China, Chengdu, China. His current research interests include fault-tolerant design of low-power AI processors and computer architecture design.

Shanshan Liu received the Ph.D. degree in Microelectronics and Solid-State Electronics from Harbin Institute of Technology, China, in 2018. She was a post-doctoral researcher at Northeastern University, Boston, USA, from 2018 to 2021, and an assistant professor at New Mexico State University, Las Cruces, USA from 2021 to 2023. She is currently a professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China.

Pedro Reviriego received the M.Sc. and Ph.D. degrees in Telecommunications engineering from the Technical University of Madrid, Madrid, Spain, in 1994 and 1997, respectively. From 1997 to 2000, he was an R&D Engineer with Teldat, Madrid. In 2000, he joined Massana and from 2004 to 2007, he was a Distinguished Member of Technical Staff with the LSI Corporation. From 2007 to 2018 he was with Universidad Antonio de Nebrija, and from 2018 to 2022 with Universidad Carlos III de Madrid. He is currently with Universidad Politécnica de Madrid.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on September 22,2026 at 08:28:13 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.